

Apuntes de Algoritmos Genéticos

Parte IV: Mutación

M. Angélica Pinninghoff J. - Ricardo Contreras A.
Magíster en Inteligencia Artificial
Universidad Adolfo Ibáñez

2022

1 Preliminares

El concepto de *especie* en biología, se entiende como aquellos miembros de una población que potencialmente son capaces de procrear; sin tener en cuenta la similitud o apariencia. Aunque la apariencia es útil en la identificación de una especie, no la determina. Existen organismos que pueden ser muy parecidos, pero que pueden corresponder a especies diferentes.

Por otra parte, organismos que se ven muy diferentes pueden pertenecer a la misma especie. Esto se da, por ejemplo, en hormigas de la especie *Pheidole barbata* en que hormigas que son *hermanas* tienen roles distintos en una misma colonia y se ven muy diferentes. Hay plantas que tienen flores rosadas o flores azules, lo que no quiere decir que debamos clasificarlas como especies diferentes. De hecho, cualquier persona podría hacer que una de estas plantas que produce flores azules, produjera en otro momento flores rosadas, simplemente modificando el pH del suelo y la cantidad de aluminio que la planta puede absorber.

En los organismos vivos se producen cambios de diferente naturaleza. Esos cambios pueden ser de comportamiento, fisiológicos, o estructurales, de manera de adecuarse de mejor forma a un ambiente determinado. Eso es lo que permite, a fin de cuentas, incrementar sus probabilidades de supervivencia. Los cambios de comportamiento se pueden ejemplificar con una transformación que permita a un animal tener mejores oportunidades de escapar de sus depredadores naturales; por ejemplo a través del camuflaje, como lo que ocurre con aquellos insectos que parecen hojas, como muestra



Figure 1: Camuflaje en un insecto

la figura 1.

Adicionalmente, en todos los seres vivos existe una forma de cambio denominada mutación, que corresponde a un cambio en la secuencia de nucleótidos de un gen o un cromosoma. La mutación es esencial para la evolución, ya que introduce nuevos rasgos en una población, incluyendo cambios en las secuencias de ADN.

En todos los seres vivos las mutaciones ocurren a baja frecuencia, debido a inestabilidades o errores que ocurren durante el proceso de replicación del ADN. La exposición natural de un organismo a ciertos factores ambientales como la radiación ultravioleta y ciertos químicos cancerígenos, también pueden provocar una mutación. Los efectos de los cambios introducidos por una mutación pueden ser heredados en las poblaciones descendientes.

2 Mutación en los algoritmos genéticos

El operador de mutación, en los algoritmos genéticos, tiene como función introducir nuevo material genético en una población, para evitar el estancamiento en las características de los individuos, y para extender el mecanismo de exploración a diferentes áreas, dentro del espacio de búsqueda [2].

Los operadores de mutación provocan cambios en las soluciones en base a modificaciones realizadas en forma random. La *fuera* de esta perturbación es lo que se conoce como tasa de mutación, y dependerá de la naturaleza del problema, aunque los valores típicos utilizados provocan cambios en un porcentaje de individuos de la población inferiores al 10%.

Hay tres requerimientos centrales para los operadores de mutación. El



Figure 2: La mutación como alteración de la información

primer requerimiento es *alcance*: cada punto en el espacio de soluciones debe ser alcanzable a partir de un punto arbitrario en ese espacio. Esta condición puede complicarse por la existencia de restricciones que reducen el espacio total de soluciones a un subconjunto *factible*. Debe existir una posibilidad, aunque sea mínima, de alcanzar cada parte del espacio de soluciones. Si esto no ocurre, puede darse que el óptimo no sea encontrado. No todos los operadores de mutación pueden garantizar esta condición.

Un segundo requerimiento para un operador de mutación es la no pre-determinación del sentido de la búsqueda; en otras palabras, el operador no debe inducir un sesgo particular en la dirección en la que se realiza la mutación. En el caso de espacios de solución restringidos, esto también puede ser una desventaja. La idea de una búsqueda novedosa, que se dirija a partes inexploradas en el espacio de soluciones, también puede introducir un sesgo en el operador de mutación.

El tercer requerimiento para un operador de mutación es la escalabilidad. Cada operador de mutación debiera ofrecer grados de libertad aceptables. Esto es posible normalmente en aquellos operadores de mutación basados en probabilidad de distribución. Por ejemplo, para la mutación Gaussiana, basada en distribución Gaussiana, la desviación estándar puede escalarse desde muestras escogidas randómicamente a todo el espacio de soluciones. En la mutación Gaussiana (normal), dado un cromosoma \mathbf{p} como j -ésimo de un gen seleccionado para mutación, se produce un cromosoma \mathbf{c} de la siguiente forma:

$$c_i = N(p, \sigma) \text{ si } j = i \\ p_i \text{ en caso contrario}$$

donde $N(p, \sigma)$ es una distribución normal con media p_i y desviación estándar σ (parámetro). De manera alternativa se puede disminuir el valor de σ a medida que aumenta el número de generaciones.

3 Tipos de Mutación

La mutación o transformación de la información contenida en un cromosoma puede tomar diversas formas de implementación [1].

Operadores simples se pueden clasificar en alguna de las siguientes categorías:

1. Mutación *textit{flipped}*. Se usa usualmente en codificaciones binarias. El cambio en este caso significa pasar del valor 1 al valor 0, o viceversa.
2. Mutación por intercambio. También se usa usualmente en codificaciones binarias. Se eligen dos posiciones aleatoriamente en un string, y los bits correspondientes a esas posiciones son intercambiados.
3. Mutación reversa. Se aplica a codificaciones binarias; se escoge de forma aleatoria una posición en el cromosoma, y luego los bits vecinos a esa posición son *flipados* (los 1s se convierten en 0s y los 0s se convierten en 1s).
4. Mutación uniforme. Se usa cuando la representación del cromosoma considera valores reales y enteros. Los valores de los genes escogidos aleatoriamente son reemplazados por valores seleccionados de un intervalo de valores posibles para ese gen.
5. Mutación *creep*. Un gen se selecciona aleatoriamente y su valor es cambiado por un valor escogido aleatoriamente en un intervalo entre un límite inferior y un límite superior.

La mutación entendida como un cambio en el contenido de información puede presentar problemas cuando se trata de problemas cuya representación trata con permutaciones. Por ejemplo, supongamos que el gen **i** tiene el valor **j** y queremos ahora cambiar ese valor de forma que ahora el gen **i** adopte el valor **k**. El cambio de algún otro valor de gen para que tenga el valor **k** podría significar que **k** está presente dos veces, y el valor **j** puede haberse perdido. En ese caso, se debe trabajar con operadores de mutación especiales.

1. Mutación por inserción. Se eligen dos valores de alelos en forma aleatoria. Luego, el segundo alelo se desplaza de manera que queda situado a continuación del primero, desplazando el resto de los alelos para reacomodarlos a continuación del recién trasladado. Esto permite mantener gran parte de la información de orden y adyacencia.
2. Mutación por inversión. Para realizar la inversión, se eligen dos alelos de manera aleatoria y luego se invierte el orden del substring que queda entre los dos alelos escogidos. Este operador preserva la mayoría de la información de adyacencia.
3. Mutación por redistribución. En este tipo de mutación se elige un subconjunto de genes de manera aleatoria, y luego se distribuyen los alelos en esas posiciones. El subconjunto elegido no requiere que los genes elegidos sean contiguos.
4. Mutación por intercambio. En esta mutación, se seleccionas dos alelos, de manera aleatoria, y se intercambian sus contenidos.

4 Ejemplos

Para entender los casos descritos arriba, incorporamos algunos ejemplos tomados de [1].

- Mutación por inserción.

Estado inicial:

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Estado final:

1	2	5	3	4	6	7	8	9
---	---	---	---	---	---	---	---	---

- Mutación por intercambio:

Estado inicial:

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Estado final:

1	5	3	4	2	6	7	8	9
---	---	---	---	---	---	---	---	---

- Mutación por inversión

Estado inicial:

1	2	3	4	5	6	7	8	9
---	----------	----------	----------	----------	---	---	---	---

Estado final:

1	5	4	3	2	6	7	8	9
---	----------	----------	----------	----------	---	---	---	---

- Mitación por redistribución

Estado inicial:

1	2	3	4	5	6	7	8	9
---	----------	----------	----------	----------	---	---	---	---

Estado final:

1	3	5	4	2	6	7	8	9
---	----------	----------	----------	----------	---	---	---	---

References

- [1] Eiben, A.E.; Smith, James E. (2003) Introduction to Evolutionary Computing. Springer.
- [2] Parmee, Ian C. (2001). Evolutionary and Adaptive Computing in Engineering Design. Springer.